

# Modeling of Heavy Gas Effects on Airfoil Flows

by

**Marc Alan Schafer**

S.B. Massachusetts Institute of Technology (1991)

SUBMITTED TO THE DEPARTMENT OF  
AERONAUTICS AND ASTRONAUTICS  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF

**Master of Science**  
at the  
**Massachusetts Institute of Technology**

June 1992

©1992, Massachusetts Institute of Technology

Signature of Author \_\_\_\_\_  
Department of Aeronautics and Astronautics  
May 3, 1992

Certified by \_\_\_\_\_  
Professor Mark Drela  
Thesis Supervisor, Department of Aeronautics and Astronautics

Accepted by \_\_\_\_\_  
Professor Harold Y. Wachman  
Chairman, Department Graduate Committee

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

JUN 12 1992

LIBRARIES

Aero



# **Modeling of Heavy Gas Effects on Airfoil Flows**

by

**Marc Alan Schafer**

Submitted to the Department of Aeronautics and Astronautics

on May 3, 1992

in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

Thermodynamic models were constructed for a calorically imperfect gas and for a non-ideal gas. These were incorporated into a quasi one dimensional flow solver to develop an understanding of the differences in flow behavior between the new models and the perfect gas model. The models were also incorporated into a two dimensional flow solver to investigate their effects on transonic airfoil flows. Specifically, the calculations simulated airfoil testing in a proposed high Reynolds number heavy-gas test facility. The results indicated that the non-idealities caused significant differences in the flow field, but that matching of an appropriate non-dimensional parameter led to flows similar to those in air.

Thesis Supervisor: **Mark Drela,**

**Associate Professor of Aeronautics and Astronautics**



## Acknowledgments

I would like to express my thanks to all those who made this thesis possible. First, to Mark Drela whose brilliance and ingenuity have served as an inspiration in all of my studies. Also, to Harold 'Guppy' Youngren whose leadership during the Daedalus project helped me to realize what it really means to be an engineer.

I would also like to thank my parents and the rest of my family. Without your support, I never would have made it as far as I have.

My appreciation also goes to the NASA Ames research center and the NDSEG fellowship program without whose financial support this thesis would never have happened.



# Contents

|   |           |
|---|-----------|
| <b>Abstract</b>   | <b>3</b>  |
| <b>Acknowledgments</b>                                  | <b>5</b>  |
| <b>1 Introduction</b>                                   | <b>10</b> |
| <b>2 Real Gases</b>                                     | <b>11</b> |
| 2.1 Calorically Imperfect Gases . . . . .               | 11        |
| 2.2 Non-Ideal Gases . . . . .                           | 12        |
| <b>3 Solving the Euler Equations</b>                    | <b>15</b> |
| 3.1 Calorically Imperfect Gas . . . . .                 | 16        |
| 3.2 Non-Ideal Gas . . . . .                             | 17        |
| <b>4 Results</b>  | <b>20</b> |
| 4.1 One Dimensional Duct Flow . . . . .                 | 21        |
| 4.2 Two Dimensional Results . . . . .                   | 23        |
| <b>5 Conclusions</b>                                    | <b>26</b> |
| <b>A Curve Fit For <math>SF_6</math> State Equation</b> | <b>27</b> |

**B MSES Subroutine for Non-Ideal Gas Model**

**28**

**Bibliography**

**43**



## List of Figures

|     |   |    |
|-----|---|----|
| 4.1 | Stagnation Pressure Ratio(Strength) vs. Upstream Mach No. for Air and $SF_6$ at 1atm and 3atm . . . . . | 20 |
| 4.2 | One dimensional Duct Flow . . . . .   | 21 |
| 4.3 | Shock Strength and Location vs. $\beta$ . . . . .   | 22 |
| 4.4 | Shock Strength and Location vs. $Z_0$ . . . . .   | 22 |
| 4.5 | Comparison of Air and $SF_6$ at Fixed $M$ and $C_L$ . . . . .   | 24 |
| 4.6 | Comparison of $SF_6$ at 1atm and 3atm to Air, $M^* = .740$ , $C_L = .9$ . . .                           | 24 |
| 4.7 | Comparison of $SF_6$ at 1atm and 3atm to Air, $M^* = .732$ , $C_L = .75$ . .                            | 25 |
| 4.8 | Comparison of $SF_6$ at 1atm and 3atm to Air, $\kappa = .439$ , $AC_L = 2.18$ . .                       | 25 |

# Chapter 1

## Introduction

In the past few decades, the design and development of large transport aircraft has relied on wind tunnel data taken at significantly lower Reynolds numbers than those found in operation. The drawbacks of this subscale data become apparent when one considers phenomena such as attachment line transition or similar aspects of boundary layer behavior at high Reynolds numbers.

The need for accurate wind tunnel data clearly mandates the construction of a suitable high Reynolds number test facility. However, the cost of building a large atmospheric tunnel and large tunnel models is prohibitive. Higher Reynolds numbers are often achieved by pressurizing tunnels to effectively increase the density of the air. This alternative is practical only up to a point.

A potential solution following the same basic idea relies upon the use of gases with significantly higher molecular weights than air. Candidate gases include Freon-12 or Sulfur Hexafluoride ( $SF_6$ ), but the use of non-breathable gases clearly causes some problems. These problems will likely be insignificant to the cost and operational advantages of such a facility. Combining heavy gases with pressurization would allow test Reynolds numbers comparable to those on large transports in flight [1].

One complication is that Freon and  $SF_6$  have significantly different thermodynamic properties than air, especially at elevated pressures. Heavy gases do not follow the ideal equation of state  $P = \rho RT$  nearly as well as air does, nor do they maintain a constant ratio of specific heats  $\gamma \equiv c_p/c_v$  over any significant temperature range. The following discussion will attempt to quantify the potential importance of these effects through a computational study.

## Chapter 2

# Real Gases

The thermodynamic relations specifically subject to real gas effects are the state equation

$$p = \rho RT \quad (2.1)$$

and the caloric equation,

$$h \equiv \int c_p dT = c_p T \quad (2.2)$$

these particular forms only being valid for a perfect gas. Real gas effects may be divided into two categories:

1. Calorically imperfect gases for which  $c_p$  depends on temperature, but which still satisfy equation (2.1).
2. Non-ideal gases for which  $c_p$  depends on both pressure and temperature, and equation (2.1) no longer holds.

The first effect results from the introduction of multiple vibrational modes for polyatomic molecules which become more important at higher temperatures. The second effect depends on intermolecular forces which become stronger as a gas moves towards liquefaction, ie. higher pressures and lower temperatures.

### 2.1 Calorically Imperfect Gases

The only difference between a perfect and an imperfect gas stems from the dependence of  $c_p$  on temperature in the imperfect case. A cursory examination of experimental data for  $SF_6$  shows that, in the range of temperatures likely to be found in a wind tunnel

test, this dependence is linear in temperature.

$$c_p(T) = a + bT \quad (2.3)$$

Therefore, equation (2.2) becomes

$$h(T) = aT + \frac{bT^2}{2} \quad (2.4)$$

which may be easily inverted to find  $T(h)$ .

$$T(h) = -\frac{a}{b} + \sqrt{\left(\frac{a}{b}\right)^2 + \frac{2h}{b}} \quad (2.5)$$

## 2.2 Non-Ideal Gases

The state equation for a perfect gas (2.1) derives from a kinetic model of gas molecules which assumes that the molecules are point masses and that they do not exert any forces on one another except instantaneously during collisions. Clearly these assumptions become less accurate as the molecular weight of the gas increases. Van der Waals's equation

$$(p + \rho^2\alpha)(1 - \rho\beta) = \rho RT \quad (2.6)$$

contains two correction to equation (2.1):  $\alpha$  corrects the pressure to account for intermolecular attraction, and  $\beta$  corrects for the volume of the molecules themselves.

Using a non-ideal state equation like Van der Waals's causes many serious complications as enthalpy,  $c_p$ ,  $\gamma$ , etc. now depend on pressure as well as temperature. Despite these complications, enthalpy and entropy must remain *state variables* regardless of the form of the state equation. That is, local entropy and enthalpy must depend only on the local pressure and temperature and *not* on the upstream conditions (ie. the gas history).

Liepmann and Roshko [2] equate this condition with the requirement that a canonical equation of state must have one of these four forms:

$$e = e(s, \rho) \quad (2.7)$$

$$h = h(s, p) \quad (2.8)$$

$$f = f(T, \rho) \quad (2.9)$$

$$g = g(T, p) \quad (2.10)$$

Here  $e = h - p/\rho$  is the usual internal energy,  $f \equiv e - Ts$  is the free energy, and  $g \equiv h - Ts$  is the free enthalpy.

For a conventional flow solver, the enthalpy definition (2.8) appears best; however, specifying the state in this specific form is not convenient because the entropy  $s$  is not readily available to the flow solver. Liepmann and Roshko propose a more suitable form

$$\frac{p}{\rho RT} = Z(p, T) \quad (2.11)$$

which requires  $T(\rho, h)$  to have a form which makes  $h$  a state variable.

For a Van der Waals's gas

$$Z = \frac{1}{1 - \beta\rho} - \frac{\alpha\rho}{RT} \quad (2.12)$$

which clearly approaches the ideal state equation for  $\alpha, \beta \rightarrow 0$ . For typically small values of  $\alpha$  and  $\beta$

$$Z \simeq 1 + \rho \left( \beta - \frac{\alpha}{RT} \right) \simeq 1 + \frac{p}{RT} \left( \beta - \frac{\alpha}{RT} \right) \quad (2.13)$$

where the second approximation is made to make  $Z = Z(p, T)$  explicitly. Liepmann and Roshko write equation (2.13) in more general form as

$$Z = 1 + \frac{p}{p_c} \phi \left( \frac{T_c}{T} \right) \quad (2.14)$$

with  $p_c$  and  $T_c$  being the critical pressure and temperature of the gas, and  $\phi$  evidently being a universal function which they tabulate for gases other than air but with approximately the same molecular weight. For heavier gases such as  $SF_6$  it is best to fit a curve to experimental data as explained in Appendix A. For  $SF_6$ , a good curve fit takes the form

$$\phi \left( \frac{T_c}{T} \right) = c_2 \left( \frac{T_c}{T} \right)^2 + c_1 \left( \frac{T_c}{T} \right) + c_0 \quad (2.15)$$

It is now necessary to determine the specific heat capacity  $c_p(p, T)$  so that the enthalpy function  $h(p, T)$  can be obtained. Liepmann and Roshko combine two forms of the

equation of state  $h(p, T)$  and  $s(p, T)$  into the fundamental reciprocity relation between  $h(p, T)$  and  $\rho(p, T)$

$$\frac{\partial h}{\partial p} = \frac{1}{\rho} - T \frac{\partial(1/\rho)}{\partial T} \quad (2.16)$$

which is valid for any gas. Combining this with the state equation (2.11) gives

$$\frac{\partial h}{\partial p} = -\frac{RT^2}{p} \left( \frac{\partial Z}{\partial T} \right)_p = \frac{RT_c}{p_c} \phi' \left( \frac{T_c}{T} \right) \equiv \mathcal{F}(T) \quad (2.17)$$

Since  $\partial h/\partial p = \mathcal{F}(T)$  only depends on the temperature, both  $h$  and  $c_p$  must be linear in the pressure as follows.

$$h(p, T) = \int \bar{c}_p(T) dT + p \mathcal{F}(T) \quad (2.18)$$

$$c_p(p, T) \equiv \frac{\partial h}{\partial T} \quad (2.19)$$

$$= \bar{c}_p(T) + p \frac{d\mathcal{F}}{dT} \quad (2.20)$$

$$= \bar{c}_p(T) - R \frac{p T_c^2}{p_c T^2} \phi'' \left( \frac{T_c}{T} \right) \quad (2.21)$$

As in the case of the calorically imperfect gas,  $\bar{c}_p(T)$  has the form

$$\bar{c}_p(T) = a + bT \quad (2.22)$$

Substituting this into the enthalpy equation gives

$$h(p, T) = aT + \frac{bT^2}{2} + \frac{pRT_c}{p_c} \phi' \left( \frac{T_c}{T} \right) \quad (2.23)$$

It is also possible to determine the caloric equation by expressing the internal energy ( $e$ ) as  $e(\rho, T)$  [3].

## Chapter 3

# Solving the Euler Equations

These gas models may be readily integrated into an existing flow solver which solves the integral form of the steady Euler equations:

$$\oint \rho \vec{u} \cdot \hat{n} dA = 0 \quad (3.1)$$

$$\oint (\rho \vec{u} \cdot \hat{n} \vec{u} + p \hat{n}) dA = 0 \quad (3.2)$$

$$h_0 \equiv h + \frac{|\vec{u}|^2}{2} = \text{constant} \quad (3.3)$$

These equations are exact for any fluid flow, but must be supplied with a state equation to relate the pressure  $p$  to the enthalpy  $h$  and the density  $\rho$ . In addition, the upwinding scheme used to capture the shocks requires the local Mach number while the boundary conditions and evaluation of shock losses require the local stagnation conditions.

It is desirable to nondimensionalize the equations, and the following scheme is used where  $\hat{()}$  denotes the dimensional quantity and  $()_{ref}$  denotes a reference quantity:

$$p = \hat{p}/p_{ref}$$

$$\rho = \hat{\rho}/\rho_{ref}$$

$$T = \hat{T}/T_{ref}$$

$$h = \frac{\hat{h} \rho_{ref}}{p_{ref}}$$

Furthermore,  $c_p$ ,  $c_v$ , and  $R$  are nondimensionalized using  $R$  resulting in several new nondimensional parameters.

$$\alpha = a/R$$

$$\beta = \frac{b T_{ref}}{2a}$$

$$\pi = P_{ref}/P_c$$

$$\tau = T_{ref}/T_c$$

For the results presented here, the reference conditions are chosen to be stagnation conditions.

### 3.1 Calorically Imperfect Gas

The nondimensional form of the caloric equation which governs the behavior of the imperfect gas is:

$$h(T) = \int c_p dT \quad (3.4)$$

$$= \alpha T + \alpha \beta T^2 \quad (3.5)$$

which may be inverted to give  $T$  as a function of  $h$ .

$$T(h) = \frac{-1 + \sqrt{1 + 4\beta h/\alpha}}{2\beta} \quad (3.6)$$

With  $T$  obtained from  $h$ ,  $p$  may be determined using the ideal gas law (2.1) and a specified value of  $\rho$ . The local Mach number comes from the familiar definition of the speed of sound:

$$a^2 \equiv \left. \frac{\partial p}{\partial \rho} \right|_s = \gamma T \quad (3.7)$$

The local value of  $\gamma$  may be found from equation (2.3).

$$\gamma = \frac{c_p}{c_v} = \frac{\alpha + 2\alpha\beta T}{1 - \alpha - 2\alpha\beta T} \quad (3.8)$$

The last remaining difficulty is the determination of the isentropic relations between pressure, density, and temperature. These relations are necessary to calculate stagnation conditions from flow conditions. The familiar perfect gas relations

$$\frac{T}{T_0} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{-1} \quad \frac{\rho}{\rho_0} = \left(\frac{T}{T_0}\right)^{\frac{1}{\gamma-1}} \quad \frac{p}{p_0} = \left(\frac{T}{T_0}\right)^{\frac{\gamma}{\gamma-1}}$$

do not hold for a calorically imperfect gas.

The proper forms are obtained from the formal statement,

$$dh = T ds + \frac{dp}{\rho} \quad (3.9)$$



and for an isentropic process  $ds = 0$ :

$$dh = \frac{dp}{\rho} \quad (3.10)$$

From the definition of enthalpy  $dh = c_p dT$ , and for an ideal gas  $p/\rho = T$ , so equation (3.10) becomes

$$\frac{c_p(T) dT}{T} = \frac{dp}{p} \quad (3.11)$$

Integrating this equation gives

$$\frac{p_0}{p} = \exp(-\alpha \log T + 2\alpha\beta(1 - T)) \quad (3.12)$$

and the isentropic density relation then follows directly from the state equation.

$$\frac{\rho}{\rho_0} = \frac{p}{p_0} \frac{T(h_0)}{T(h)} \quad (3.13)$$

Strictly speaking, solution of the Euler equations requires nothing else. However, if a Newton-Raphson technique is used, all of the necessary equations must be linearized for the Jacobian matrix. In the case of the calorically imperfect gas, the equations are slightly more complicated than for a perfect gas, but they may still all be written explicitly. Therefore the linearizations are easily done by differentiating the relevant equations.

## 3.2 Non-Ideal Gas

The nondimensional equations describing the non-ideal gas are the state equation

$$\frac{p}{\rho T} = \frac{1 + p\pi\phi(\frac{1}{\tau T})}{Z_0} \quad (3.14)$$

and the caloric equation.

$$h(p, T) = \left[ \alpha T + \alpha\beta T^2 + p \frac{\pi}{\tau} \phi'(\frac{1}{\tau T}) \right] \frac{1}{Z_0} \quad (3.15)$$

$Z_0$  is another parameter which may be described in terms of  $\pi$  and  $\tau$ .

$$Z_0 = \frac{p_0}{\rho_0 T_0} = 1 + p_0 \pi \phi(\frac{1}{\tau T_0}) \quad (3.16)$$

The non-ideal gas presents some difficulty as the enthalpy depends on the temperature *and* the pressure. Therefore, from equations (3.14) and (3.15),  $p$  and  $T$  may be found using a Newton-Raphson system to drive the following residuals to zero.

$$R_1(p, T) = \frac{p}{\rho T} - \frac{1 + p\pi\phi(\frac{1}{\tau T})}{Z_0} \quad (3.17)$$

$$R_2(p, T) = h - \left[ \alpha T + \alpha\beta T^2 + p\frac{\pi}{\tau}\phi'(\frac{1}{\tau T}) \right] \frac{1}{Z_0} \quad (3.18)$$

The local Mach number depends on the speed of sound which must be found from the definition:

$$a^2 = \left. \frac{\partial p}{\partial \rho} \right|_s \quad (3.19)$$

This is calculated as follows:

$$dp = \left. \frac{\partial p}{\partial \rho} \right|_h d\rho + \left. \frac{\partial p}{\partial h} \right|_\rho dh \quad (3.20)$$

but  $dh = dp/\rho$  for an isentropic process, and hence

$$a^2 = \left. \frac{\partial p}{\partial \rho} \right|_s = \frac{\left. \frac{\partial p}{\partial \rho} \right|_h}{1 - \left. \frac{\partial p}{\partial h} \right|_\rho \frac{1}{\rho}} \quad (3.21)$$

The local  $\gamma$  really has no meaning and need not be calculated.

The extra complexity of the non-ideal gas appears in the calculation of the sensitivities. Since  $p$  and  $T$  are found by an iterative process they must be found by perturbing the Jacobian matrix of the converged Newton-Raphson system. A perturbation in  $h$  and  $\rho$  is related to a perturbation in  $p$  and  $T$  by the condition that the  $R(p, T, h, \rho)$  must remain zero.

$$\begin{Bmatrix} \delta R_1 \\ \delta R_2 \end{Bmatrix} = 0 = \begin{bmatrix} \frac{\partial R_1}{\partial h} & \frac{\partial R_1}{\partial \rho} \\ \frac{\partial R_2}{\partial h} & \frac{\partial R_2}{\partial \rho} \end{bmatrix} \begin{Bmatrix} \delta h \\ \delta \rho \end{Bmatrix} + \begin{bmatrix} \frac{\partial R_1}{\partial p} & \frac{\partial R_1}{\partial T} \\ \frac{\partial R_2}{\partial p} & \frac{\partial R_2}{\partial T} \end{bmatrix} \begin{Bmatrix} \delta p \\ \delta T \end{Bmatrix} \quad (3.22)$$

Numerically inverting this system gives the required derivatives.

$$\begin{Bmatrix} \delta p \\ \delta T \end{Bmatrix} = \begin{bmatrix} \frac{\partial p}{\partial h} & \frac{\partial p}{\partial \rho} \\ \frac{\partial T}{\partial h} & \frac{\partial T}{\partial \rho} \end{bmatrix} \begin{Bmatrix} \delta h \\ \delta \rho \end{Bmatrix} \quad (3.23)$$

The second derivatives are found in a similar fashion starting instead with  $\frac{\partial R_n}{\partial h}$  and  $\frac{\partial R_n}{\partial \rho}$  as the residuals. Using a subscript notation for the derivatives ( $\frac{\partial p}{\partial h} \equiv p_h$ ):

$$\begin{Bmatrix} \delta R_{1h} \\ \delta R_{2h} \end{Bmatrix} = 0 = \begin{bmatrix} \frac{\partial R_{1h}}{\partial h} & \frac{\partial R_{1h}}{\partial \rho} \\ \frac{\partial R_{2h}}{\partial h} & \frac{\partial R_{2h}}{\partial \rho} \end{bmatrix} \begin{Bmatrix} \delta h \\ \delta \rho \end{Bmatrix} + \begin{bmatrix} \frac{\partial R_{1h}}{\partial p_h} & \frac{\partial R_{1h}}{\partial T_h} \\ \frac{\partial R_{2h}}{\partial p_h} & \frac{\partial R_{2h}}{\partial T_h} \end{bmatrix} \begin{Bmatrix} \delta p_h \\ \delta T_h \end{Bmatrix} \quad (3.24)$$

A similar system with  $R_{1\rho}$  and  $R_{2\rho}$  as residuals is also formed. As above, numerically inverting gives  $\frac{\partial^2 p}{\partial h^2} = \frac{\partial p_h}{\partial h}$ ,  $\frac{\partial^2 p}{\partial h \partial \rho} = \frac{\partial p_h}{\partial \rho}$ , etc. These manipulations are implemented in the source code in Appendix B.

The last remaining task is calculation of the stagnation conditions and, again, it is not possible to find an analytic expression. Another Newton-Raphson system is constructed where the first residual comes from equation(3.15):

$$R_1 = h_0 - h(p, T) \quad (3.25)$$

The second residual is derived by rearranging equation(3.9)

$$ds = \frac{dh}{T} + \frac{dp}{\rho T} \quad (3.26)$$

$$= \frac{\bar{c}_p}{T} dT + \frac{d(p\mathcal{F})}{T} - \frac{dp}{p} Z \quad (3.27)$$

$$= \frac{\bar{c}_p}{T} dT + d\left(p\pi \frac{1}{\tau T} \phi'\right) - \pi d(p\phi) - \frac{dp}{p} \quad (3.28)$$

Integrating gives:

$$s(p, T) = \int \frac{\bar{c}_p}{T} dT + p\pi \left[ \frac{1}{\tau T} \phi' - \phi \right] - \ln(p) \quad (3.29)$$

The second residual may then be formed

$$R_2 = s_1 - s(p, T) \quad (3.30)$$

where  $s_1$  is the entropy of the static conditions.

Driving these two residuals to zero gives the stagnation conditions  $p_0, T_0$ . The derivatives  $\frac{\partial p_0}{\partial \rho}$ ,  $\frac{\partial p_0}{\partial h}$ , etc, needed for the Newton-Raphson solver may then be found by perturbing the converged Jacobian matrix and relating the resulting derivatives to the static conditions through the chain rule and equations (3.15) and (3.29). This process is identical to the one used above to find  $p$  and  $T$  and their derivatives.

## Chapter 4

### Results

After developing the models for the calorically imperfect and non-ideal gases, the next step was to evaluate the differences these changes caused in inviscid flows. The primary quantities of interest are the location of shocks and their strength which is defined as the ratio of of stagnation pressures across the shock. For a perfect gas, the shock strength may be expressed as a function of the upstream Mach number  $M_1$ .

$$\frac{p_{02}}{p_{01}} = \left[ 1 + \frac{2\gamma}{\gamma+1}(M_1^2 - 1) \right]^{-1/(\gamma-1)} \left[ \frac{(\gamma+1)M_1^2}{(\gamma-1)M_1^2 + 2} \right]^{\gamma/(\gamma-1)} \quad (4.1)$$

However, for the non-ideal gas, this relation must be calculated numerically.

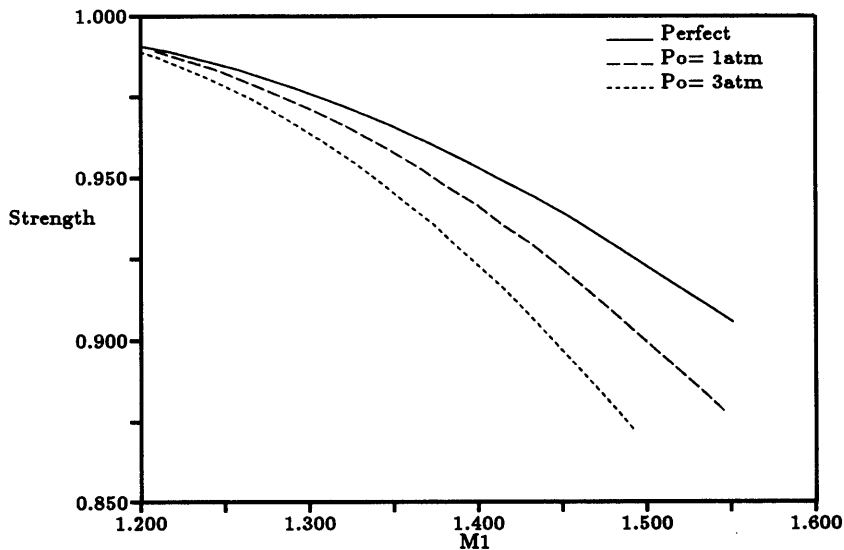


Figure 4.1: Stagnation Pressure Ratio(Strength) vs. Upstream Mach No. for Air and  $SF_6$  at 1atm and 3atm

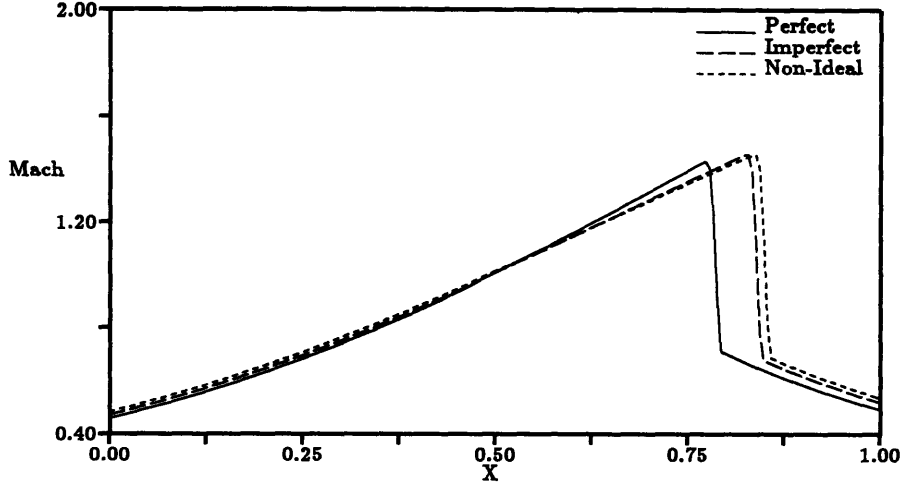


Figure 4.2: One dimensional Duct Flow

## 4.1 One Dimensional Duct Flow

The first comparison of the different gas models was a study of the flow in a converging/diverging nozzle using a quasi one dimensional Euler solver. This flow is characterized by sonic flow at the throat with a shock downstream to match the specified exit pressure as shown in figure(4.2).

As a basis for comparison of the different gas models in a duct flow, the non-dimensional reference enthalpy ( $h_0\rho_0/p_0$ ) was made equal for all three cases.

$$h_0 = \frac{\gamma}{\gamma - 1} \quad (4.2)$$

$$= \alpha(1 + \beta) \quad (4.3)$$

$$= \frac{\alpha(1 + \beta) + \pi\phi'(\frac{1}{\tau})}{Z_0} \quad (4.4)$$

With  $h_0$  held constant,  $\gamma$  therefore depends on  $\alpha$ ,  $\beta$ ,  $\pi$ , and  $\tau$ . The exit pressure ratio is also held constant. Under these conditions, the slope of the  $c_p$  versus  $T$  curve ( $\beta$ ) had little or no effect on shock strength or position relative to the perfect gas as shown in figure(4.3).

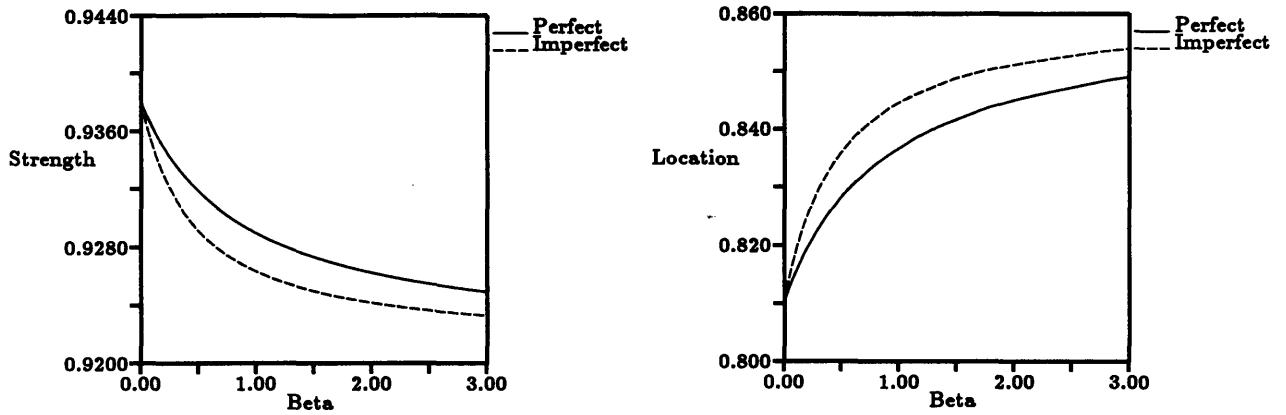


Figure 4.3: Shock Strength and Location vs.  $\beta$

For the non-ideal gas,  $\pi$  and  $\tau$  are not really independent parameters and may be combined into  $Z_0$ . Figure(4.4) shows the variation in shock strength and position as functions of  $Z_0$  and the corresponding perfect gas results with  $\gamma$  adjusted to preserve the stagnation enthalpy as above. These plots clearly show that it is not possible to mimic the effects of the non-ideality by changing  $\gamma$  as in the case of the calorically imperfect gas. The difference in shock strength and position becomes larger and larger as the gas becomes less ideal.

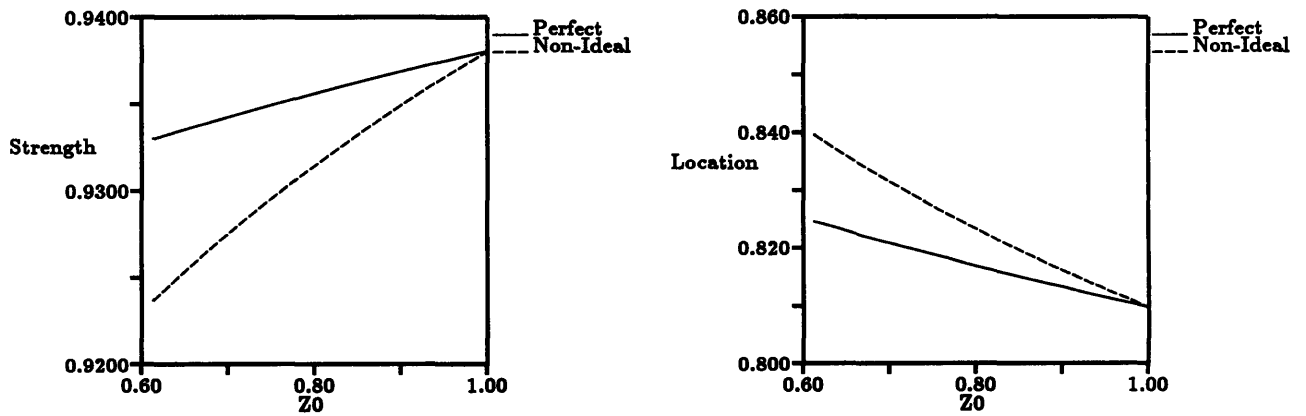


Figure 4.4: Shock Strength and Location vs.  $Z_0$

The last test conducted with the one dimensional flow model was to determine the effects of the various gas models on the upwinding scheme needed for stability of the numerical scheme. The flow solver drives the momentum equation residual to zero,

$$R_1 \equiv \rho_i q_i A_i (\bar{q}_i - q_{i-1}) + p_i A_i - p_{i-1} A_{i-1} + \frac{p_i + p_{i-1}}{2} (A_i - A_{i-1}) \quad (4.5)$$

where the upwinded speed is defined as

$$\bar{q}_i = q_i - \mu_i (q_i - q_{i-1}) \quad (4.6)$$

and  $\mu_i$  is non-zero only if  $M_i$  is greater than  $M_c$ .

$$\mu_i(M_i(q_i)) = \frac{K_\mu}{\gamma} \left[ 1 - \frac{M_c^2}{M_i^2} \right] \quad (4.7)$$

Initially, the exact  $\gamma$  was calculated at each node along with all the necessary linearizations and used in the upwinding scheme. Under these conditions, the flow solver converged with  $M_c \leq 1$ . However, the upwinding is relatively insensitive to the exact value of  $\gamma$  even though the stability analysis used to derive equation(4.7) ignored  $\gamma$  perturbations. Using a constant value of  $\gamma$  had absolutely no effect on the viable range for  $M_c$  or the rate of convergence.

## 4.2 Two Dimensional Results

The subroutine which appears in Appendix B was incorporated into MSES, the multi-element version of the two dimensional transonic airfoil design/analysis code ISES [4]. Numerical experiments carried out were limited to single-element inviscid cases to more clearly demonstrate the effect of the new gas model. Figure(4.5) shows an overlay of the Mach distributions for a test airfoil run in  $SF_6$  at two different stagnation conditions and in air. All three cases are at matched freestream Mach number and lift coefficient. Note that they are *not* at the same angle of attack. The  $SF_6$  is characterized by stagnation pressures of 1atm and 3atm and a stagnation temperature of 310K.

Airfoils tests in heavy gases will be much more worthwhile if some relationship may be found so that the tests reflect the airfoil performance in air. The only parameters

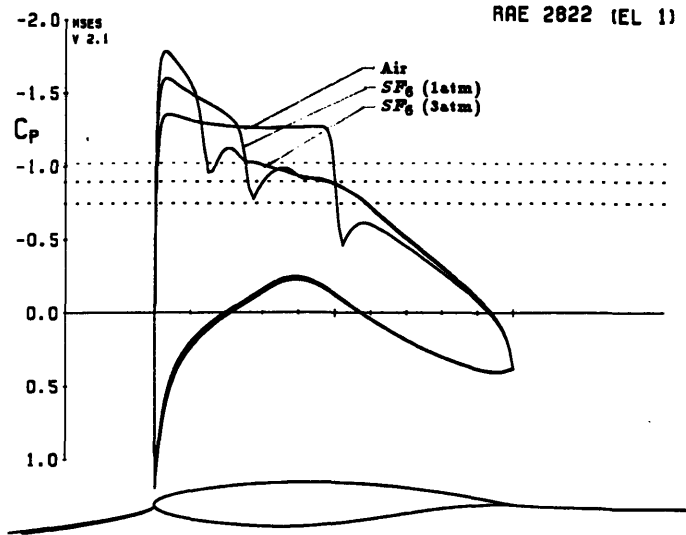


Figure 4.5: Comparison of Air and  $SF_6$  at Fixed  $M$  and  $C_L$

which may be adjusted in a wind tunnel test are the Mach number, stagnation conditions, and angle of attack or  $C_L$ . Figure(4.5) shows an attempted match keeping  $M$  and  $C_L$  constant: clearly, this is not an effective technique. After a good deal of experimentation, the best match was achieved by running the different gases at the same  $M^*$  which is defined as the ratio of freestream velocity to the speed of sound at sonic conditions. Figure(4.6) shows the case in air from figure(4.5) compared with  $SF_6$  (1atm and 3atm) at the same  $M^*$ .

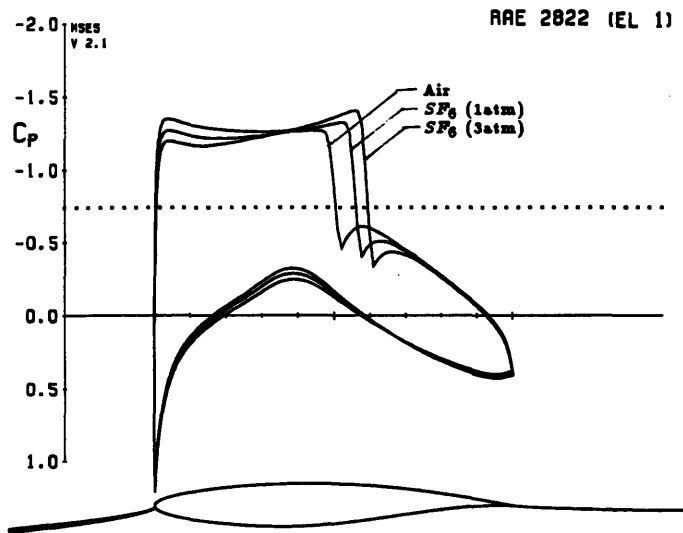


Figure 4.6: Comparison of  $SF_6$  at 1atm and 3atm to Air,  $M^* = .740$ ,  $C_L = .9$



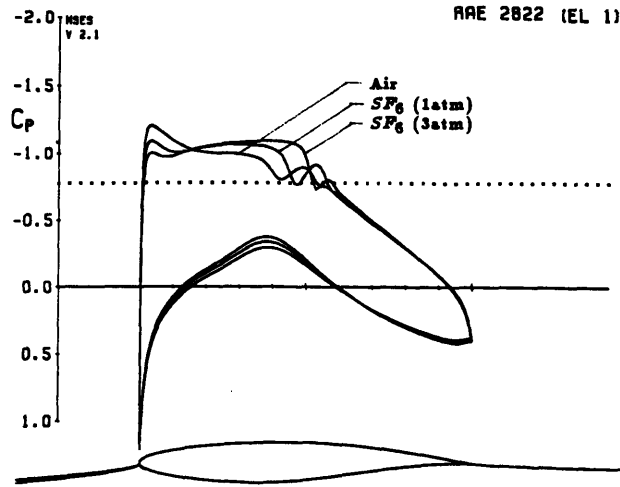


Figure 4.7: Comparison of  $SF_6$  at 1atm and 3atm to Air,  $M^* = .732$ ,  $C_L = .75$

A case with a weaker shock, figure(4.7) was used to further verify this relationship. The match is slightly worse, but this is to be expected because a weak shock is much more sensitive to small changes in  $M$  than a strong one. As an alternative to matching  $M^*$ , Anderson [5] proposes matching the small disturbance similarity parameter  $\kappa$  and  $AC_L$  where

$$\kappa = \frac{1 - M_\infty^2}{(M_\infty^2(\gamma' + 1))^{2/3}} \quad (4.8)$$

$$A = \frac{M_\infty^2(\gamma' + 1)}{1 - M_\infty^2} \quad (4.9)$$

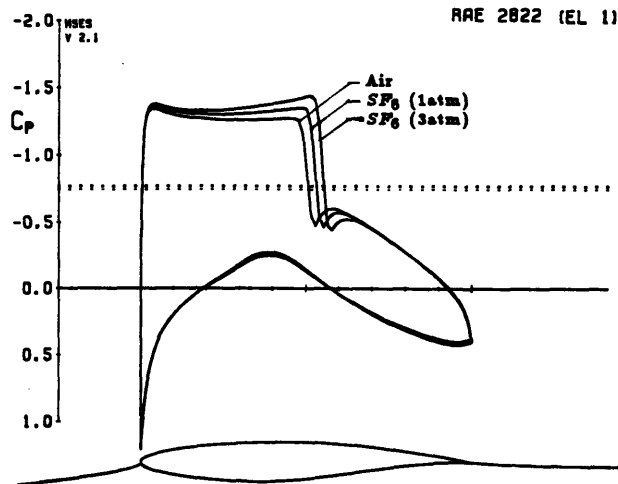


Figure 4.8: Comparison of  $SF_6$  at 1atm and 3atm to Air,  $\kappa = .439$ ,  $AC_L = 2.18$

## Chapter 5

# Conclusions

The models derived above adequately describe the thermodynamic behavior of non-ideal and calorically imperfect gases. Despite some minor complications in linearizing these models, they were implemented in routines suitable for incorporation into existing flow solvers based on Newton's method. First, a quasi one-dimensional flow solver was used to examine the influence of the various non-dimensional parameters which govern the behavior of the different gases.

Transonic airfoil test cases for air and  $SF_6$  were then used to study the influence of parameters which may be controlled in a wind tunnel experiment: stagnation pressure, freestream Mach number, and angle of attack. The goal of this study was determine the conditions under which a wind tunnel test in a heavy gas would produce results comparable to those found in air. Matching  $M^*$  and  $C_L$  or  $\kappa$  and  $AC_L$  were both effective for the test cases presented here. Further study is necessary to determine which is best for multi-element cases.

The results are encouraging in that they definitely hint at the possibility of directly relating heavy gas test data to performance in air. It is first necessary to verify experimentally the model for  $SF_6$ , and to investigate the effects of non-ideal gases on viscous flows.

## Appendix A

### Curve Fit For $SF_6$ State Equation

A curve fit may be found for the function  $\phi\left(\frac{T}{T^*}\right)$  for any gas given experimental state data. With the density ( $\rho$ ) measured at a number of different pressures ( $p$ ) and temperatures ( $T$ ), a vector is defined containing the difference between the real gas and a perfect gas at each data point.

$$\vec{Z} = \begin{bmatrix} \frac{p_1}{\rho_1 RT_1} - 1 \\ \vdots \\ \frac{p_m}{\rho_m RT_m} - 1 \end{bmatrix} \quad (\text{A.1})$$

Defining  $\theta \equiv \frac{T}{T^*}$ , the matrix  $A$  contains the state information.

$$\bar{A} = \frac{1}{p_{ref}} \begin{bmatrix} p_1 \theta_1^n & p_1 \theta_1^{n-1} & \dots & p_1 \theta_1^2 & -p_1 \theta_1 & p_1 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ p_m \theta_m^n & p_m \theta_m^{n-1} & \dots & p_m \theta_m^2 & p_m \theta_m & p_m \end{bmatrix} \quad (\text{A.2})$$

The goal is to find a state equation agreeing closely with the experimental data in  $\vec{Z}$  but of the simple form:

$$Z(p, T) = 1 + \frac{p}{p_{ref}} \begin{bmatrix} C_n & C_{n-1} & \dots & C_0 \end{bmatrix} \begin{bmatrix} \theta^n \\ \theta^{n-1} \\ \vdots \\ 1 \end{bmatrix} \quad (\text{A.3})$$

Therefore

$$\vec{Z} \simeq A \vec{C} \quad (\text{A.4})$$

and  $\vec{C}$  is found by the technique of linear regression:

$$\vec{C} = (A^T A)^{-1} A^T Z \quad (\text{A.5})$$

The results presented in this thesis were based on a quadratic fit for  $\phi$  from approximate data for  $SF_6$ . The required data may be found in [6].

## Appendix B

### MSES Subroutine for Non-Ideal Gas Model

```

subroutine hgparm(alf1,btai, tau1, cc0,cc1,cc2, h0)
-----
c   Initializes non-ideal gas routines.
c   Formulation derived in Schafer SM thesis.
c
c   Input:
c   alf1  Constants for Cp(T) in caloric equation: Cp = a(1 + bT)
c   btai
c
c   tau1  Constant in phi(T) in non-ideality factor Z(p,T)
c
c   cc0   Constants defining phi(T) in polynomial form:
c   cc1
c   cc2   phi = c0 + c1(tau/T) + c2(tau/T)**2
c
c   Output:
c   h0    Enthalpy at reference conditions p0, T0
c
c   Internal output:
c   z0    Non-ideality factor Z(p0,T0) at reference conditions
c
-----
      implicit real*4 (a-h,m,o-z)
      common /nongas/
      &      alf, bta, pi, tau, z0
      common /nonfit/
      &      c2, c1, c0
c
c----- put input parameters into common blocks
      alf = alf1
      bta = btai
c
      tau = tau1
c
      c0 = cc0
      c1 = cc1
      c2 = cc2
c
      pi = 1.0
c
c----- calculate reference non-ideality factor and enthalpy
      z0 = 1.0 + pi*(c2/tau**2 + c1/tau + c0)
      h0 = (alf*(1. + bta) + pi/tau*phid(1./tau)) / z0
c

```

```

return
end

subroutine nideal(h0,r,q, p ,p_r ,p_q,
&               msq,msq_r,msq_q)
c-----
c   Calculates pressure and Mach number for specified
c   stagnation enthalpy, density, and speed.
c
c   Input:
c   h0      stagnation enthalpy
c   r       density
c   q       speed
c
c   Output:
c   p       pressure
c   p_r     dp/dr
c   p_q     dp/dq
c   msq     square of Mach number M^2
c   msq_r   dM^2/dr
c   msq_q   dM^2/dq
c-----
      implicit real*4 (a-h,m,o-z)
c
c---- set static enthalpy
      h = h0 - 0.5*q**2
      h_q =      -q
c
c---- set pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
&               t,t_r,t_h,t_rr,t_hh,t_rh)
      p_q = p_h*h_q
c
c---- set speed of sound squared:  a^2 = dp/dr (at constant s)
      asq = p_r / (1. - p_h/r)
      asq_r = p_rr / (1. - p_h/r)
&      - p_r / (1. - p_h/r)**2 *(p_h/r**2 - p_rh/r)
      asq_h = p_rh / (1. - p_h/r)
&      + p_r / (1. - p_h/r)**2 *p_hh/r
      asq_q = asq_h*h_q
c
c---- set Mach number squared
      msq = q**2/asq
      msq_r = -msq/asq * asq_r
      msq_q = -msq/asq * asq_q + 2.*q/asq
c
return
end

subroutine ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,

```

```

      &          t,t_r,t_h,t_rr,t_hh,t_rh)
c-----
c   Calculates pressure and temperature for
c   specified static enthalpy and density.
c
c   Input:
c   h      enthalpy
c   r      density
c
c   Output:
c   p      pressure
c   p_r    dp/dr
c   p_h    dp/dh
c   p_rr   d^2p/dr^2
c   p_hh   d^2p/dh^2
c   p_rh   d^2p/drdh
c   t      temperature
c   t_r    dt/dr ... etc.
c-----
      implicit real*4 (a-h,m,o-z)
      dimension a(2,2), ai(2,2), aih(2,2), air(2,2),
& b(2,2), bh(2,2), br(2,2)
      common /nongas/
&          alf, bta, pi, tau, z0
c
c----- Newton convergence tolerance
      data eps /5.0E-6/
c
c----- initial guess from imperfect ideal gas
      if(bta.eq.0.0) then
          t = h/alf
      else
          t = (-1.0 + sqrt(1.0 + 4.0*bta*h/alf)) / (2.0*bta)
      endif
      p = r*t
c
c----- Newton loop to converge on correct p,t
      itcon = 15
      do 100 iter=1, itcon
c
c----- set and linearize non-ideality factor Z(p,t)
          ttc = 1./(tau*t)
          ttc_t = -1./(tau*t**2)
c
          z = 1. + p*pi*phi(ttc)
          z_p = pi*phi(ttc)
          z_t = p*pi*phid(ttc)*ttc_t
c
c----- residual 1: state equation
          res1 = p/(r*t) - z /z0
          r1_p = 1./(r*t) - z_p/z0
          r1_t = -p/(r*t**2) - z_t/z0
c
c

```

```

      tm1 = (alf*t +   alf*bta*t**2) / z0
      tm1_p = 0.
      tm1_t = (alf   + 2.*alf*bta*t   ) / z0
c
      tm2 = p*pi/tau*phid(ttc)          / z0
      tm2_p = pi/tau*phid(ttc)          / z0
      tm2_t = p*pi/tau*phidd(ttc)*ttc_t / z0
c
c---- residual 2: caloric equation
      res2 = h - (tm1 + tm2)
      r2_p = - (tm1_p + tm2_p)
      r2_t = - (tm1_t + tm2_t)
c
c---- set Jacobian matrix
      a(1,1) = r1_t
      a(1,2) = r1_p
      a(2,1) = r2_t
      a(2,2) = r2_p
c
c---- find inverse Jacobian matrix
      detinv = 1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
      ai(1,1) = a(2,2)*detinv
      ai(2,2) = a(1,1)*detinv
      ai(1,2) = -a(1,2)*detinv
      ai(2,1) = -a(2,1)*detinv
c
c---- set Newton changes
      dt = -(ai(1,1)*res1 + ai(1,2)*res2)
      dp = -(ai(2,1)*res1 + ai(2,2)*res2)
c
      rlx = 1.0
      if(rlx*dp .gt. 2.5*p) rlx = 2.5*p/dp
      if(rlx*dp .lt. -.8*p) rlx = -.8*p/dp
      if(rlx*dt .gt. 2.5*t) rlx = 2.5*t/dt
      if(rlx*dt .lt. -.8*t) rlx = -.8*t/dt
c
c---- update variables
      t = t + rlx*dt
      p = p + rlx*dp
c
c---- convergence check
      if (abs(dp/p) .le. eps .and. abs(dt/t) .le. eps) goto 3
c
100 continue
c
      write(*,*) 'NGASPT: Convergence failed.'
      write(*,*) 'dp dT :', dp, dt
      write(*,*) 'p T h r:', p, t, h, r
c
3 continue
c
c---- set residual derivatives wrt input r,h variables
      r1_r = -p/(r**2*t)
      r1_h = 0.

```

```

r2_r = 0.
r2_h = 1.
c
b(1,1) = r1_r
b(1,2) = r1_h
b(2,1) = r2_r
b(2,2) = r2_h
c
c---- set p,t derivatives wrt r,h
t_r = -(ai(1,1)*b(1,1) + ai(1,2)*b(2,1))
t_h = -(ai(1,1)*b(1,2) + ai(1,2)*b(2,2))
p_r = -(ai(2,1)*b(1,1) + ai(2,2)*b(2,1))
p_h = -(ai(2,1)*b(1,2) + ai(2,2)*b(2,2))
c
c
c---- set second residual derivatives wrt r,h
ttc = 1./(tau*t)
ttc_t = -1./(tau*t**2)
ttc_tt = 2./(tau*t**3)
c
z = 1. + p*pi*phi(ttc)
z_p = pi*phi(ttc)
z_pt = pi*phid(ttc)*ttc_t
z_pp = 0.
z_t = p*pi*phid(ttc)*ttc_t
z_tt = p*pi*(phidd(ttc)*ttc_t**2 + phid(ttc)*ttc_tt)
c
r1 = p/(r*t) - z /z0
r1_p = 1./(r*t) - z_p /z0
r1_pt = -1./(r*t**2) - z_pt/z0
r1_pp = - z_pp/z0
r1_t = -p/(r*t**2) - z_t /z0
r1_tt = 2.*p/(r*t**3) - z_tt/z0
r1_r = -p/(r**2*t)
r1_h = 0.
r1_hp = 0.
r1_ht = 0.
r1_rp = -1./(r**2*t)
r1_rt = p/(r**2*t**2)
r1_rr = 2.*p/(r**3*t)
c
tm1 = (alf*t + alf*bta*t**2) / z0
tm1_t = (alf + 2.*alf*bta*t) / z0
tm1_tt = ( 2.*alf*bta ) / z0
tm1_pt = 0.
tm1_p = 0.
tm1_pp = 0.
c
tm2 = p*pi/tau*phid(ttc) / z0
tm2_p = pi/tau*phid(ttc) / z0
tm2_pt = pi/tau*phidd(ttc)*ttc_t / z0
tm2_pp = 0.
tm2_t = p*pi/tau* phidd(ttc)*ttc_t / z0
tm2_tt = p*pi/tau*(phidd(ttc)*ttc_t**2 +

```



```

*          phidd(ttc)*ttc_tt)      / z0
c
r2   = h - (tm1  + tm2)
r2_p = - (tm1_p + tm2_p)
r2_t = - (tm1_t + tm2_t)
r2_h = 1.
c
c
c---- set and linearize new residuals: r1h = dr1/dh = 0, r2h = dr2/dh = 0
ph = p_h
th = t_h
c
r1h   = r1_p *ph + r1_t *th + r1_h
r1h_ph = r1_p
r1h_th =          r1_t
r1h_p  = r1_pp*ph + r1_pt*th + r1_hp
r1h_t  = r1_pt*ph + r1_tt*th + r1_ht
r1h_h  = 0.
r1h_r  = -ph/(r**2*t) + th*p/(r**2*t**2)
c
c
r2h   = 1. - tm1_t*th - tm1_p*ph - tm2_t*th - tm2_p*ph
r2h_ph =          - tm1_p          - tm2_p
r2h_th = - tm1_t          - tm2_t
r2h_p  = - tm1_pt*th - tm1_pp*ph - tm2_pt*th - tm2_pp*ph
r2h_t  = - tm1_tt*th - tm1_pt*ph - tm2_tt*th - tm2_pt*ph
r2h_h  = 0.
r2h_r  = 0.
c
a(1,1) = r1h_th
a(1,2) = r1h_ph
a(2,1) = r2h_th
a(2,2) = r2h_ph
c
detinv = 1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
aih(1,1) = a(2,2)*detinv
aih(2,2) = a(1,1)*detinv
aih(1,2) = -a(1,2)*detinv
aih(2,1) = -a(2,1)*detinv
c
dth = -(aih(1,1)*r1h + aih(1,2)*r2h)
dph = -(aih(2,1)*r1h + aih(2,2)*r2h)
c
c   ph = ph + dph
c   th = th + dth
c
c
c---- set and linearize new residuals: r1r = dr1/dr = 0, r2r = dr2/dr = 0
pr = p_r
tr = t_r
c
r1r   = r1_p *pr + r1_t *tr + r1_r
r1r_pr = r1_p
r1r_tr =          r1_t

```

```

r1r_p = r1_pp*pr + r1_pt*tr + r1_rp
r1r_t = r1_pt*pr + r1_tt*tr + r1_rt
r1r_r = r1_rp*pr + r1_rt*tr + r1_rr
r1r_h = 0.

c
c
r2r    = - tm1_t *tr - tm1_p *pr - tm2_t *tr - tm2_p *pr
r2r_pr =          - tm1_p          - tm2_p
r2r_tr = - tm1_t          - tm2_t
r2r_p  = - tm1_pt*tr - tm1_pp*pr - tm2_pt*tr - tm2_pp*pr
r2r_t  = - tm1_tt*tr - tm1_pt*pr - tm2_tt*tr - tm2_pt*pr
r2r_h  = 0.
r2r_r  = 0.

c
a(1,1) = r1r_tr
a(1,2) = r1r_pr
a(2,1) = r2r_tr
a(2,2) = r2r_pr

c
detinv = 1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
air(1,1) = a(2,2)*detinv
air(2,2) = a(1,1)*detinv
air(1,2) = -a(1,2)*detinv
air(2,1) = -a(2,1)*detinv

c
dtr = -(air(1,1)*r1r + air(1,2)*r2r)
dpr = -(air(2,1)*r1r + air(2,2)*r2r)

c
c
pr = pr + dpr
tr = tr + dtr
c
c
c---- calculate responses in dt/dh and dp/dh to unit h perturbation
dr1h = r1h_h + r1h_p*ph + r1h_t*th
dr2h = r2h_h + r2h_p*ph + r2h_t*th

c
dr1r = r1r_h + r1r_p*ph + r1r_t*th
dr2r = r2r_h + r2r_p*ph + r2r_t*th

c
c
dth = -(aih(1,1)*dr1h + aih(1,2)*dr2h)
dph = -(aih(2,1)*dr1h + aih(2,2)*dr2h)
thh = dth
phh = dph

c
dth = -(air(1,1)*dr1r + air(1,2)*dr2r)
dph = -(air(2,1)*dr1r + air(2,2)*dr2r)
thr = dth
phr = dph

c
c---- calculate responses in dt/dh and dp/dh to unit r perturbation
dr1h = r1h_r + r1h_p*pr + r1h_t*tr
dr2h = r2h_r + r2h_p*pr + r2h_t*tr

c

```

```

dr1r = r1r_r + r1r_p*pr + r1r_t*tr
dr2r = r2r_r + r2r_p*pr + r2r_t*tr
c
c
dth = -(aih(1,1)*dr1h + aih(1,2)*dr2h)
dph = -(aih(2,1)*dr1h + aih(2,2)*dr2h)
trh = dth
prh = dph
c
dth = -(air(1,1)*dr1r + air(1,2)*dr2r)
dph = -(air(2,1)*dr1r + air(2,2)*dr2r)
trr = dth
prr = dph
c
c---- set final first and second derivatives wrt (r,h)
p_r = pr
t_r = tr
p_h = ph
t_h = th
p_hh = phh
t_hh = thh
p_rr = prr
t_rr = trr
p_rh = .5*(prh+phr)
t_rh = .5*(trh+thr)
c
return
end

subroutine nonstag(h0,rho,q, p0,p0_r,p0_q,
& r0,r0_r,r0_q )
c-----
c
c Calculates stagnation pressure and density for
c specified stagnation enthalpy, density, and speed.
c
c Input:
c h0 stagnation enthalpy
c rho density
c q speed
c
c Output:
c p0 stagnation pressure
c p0_r dp0/dr
c p0_q dp0/dq
c r0 stagnation density
c r0_r dr0/dr
c r0_q dr0/dq
c-----
c
c implicit real*4 (a-h,m,o-z)
c dimension a(2,2), ai(2,2), b(2,2)
c real*4 h_p,h_t
c

```

```

common /nongas/
&      alf, bta, pi, tau, z0
common /nonfit/
&      c2, c1, c0

c
data eps /5.0E-6/

c
z(pp,tt) = 1. + pp*pi*phi (1./(tau*tt))
z_p(pp,tt) =      pi*phi (1./(tau*tt))
z_t(pp,tt) =      pp*pi*phid(1./(tau*tt)) / (-tau*tt**2)

c
h      = h0 - .5*q**2
h_q    =      - q
ccc    h_h0 = 1.0

c
r = rho

c
c---- set input pressure and temperature and derivatives
call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
&          t,t_r,t_h,t_rr,t_hh,t_rh)

c
c---- set entropy s and derivatives wrt p,t
ttc    = 1./(tau*t)
ttc_t  = -1./(tau*t**2)
ttc_tt = 2./(tau*t**3)

c
ph     = phi(ttc)
phd    = phid(ttc)
phdd   = phidd(ttc)
phddd  = phidd(ttc)

c
ph_t   = phd  * ttc_t
phd_t  = phdd * ttc_t
phdd_t = phddd * ttc_t

c
s      = alf*log(t) + 2.0*alf*bta*t
&      - p*pi*( t*phd  *ttc_t + ph ) - log(p)
s_p    = - pi*( t*phd  *ttc_t + ph ) - 1.0/p
s_t    = alf/t      + 2.0*alf*bta
&      - p*pi*(  phd  *ttc_t + ph_t
&                + t*phd_t*ttc_t
&                + t*phd  *ttc_tt )

c
c---- initial guess for p0,t0 from imperfect gas
cc     if(bta.eq.0.0) then
cc       t0 = h0/alf
cc     else
cc       t0 = (-1.0 + sqrt(1.0 + 4.0*bta*h0/alf)) / (2.0*bta)
cc     endif
cc     p0 = p * exp(-alf*log(t) + alf*2.0*bta*(1.0-t))

c
t0 = t
p0 = p

c

```

```

c---- Newton loop to converge on correct p0,t0
itcon = 15
do 100 iter=1, itcon
c
  ttc      = 1./(tau*t0)
  ttc_t0   = -1./(tau*t0**2)
  ttc_tt0  = 2./(tau*t0**3)
c
  ph       = phi(ttc)
  phd      = phid(ttc)
  phdd     = phidd(ttc)
  phddd    = phiddd(ttc)
c
  ph_t0    = phd   * ttc_t0
  phd_t0   = phdd  * ttc_t0
  phdd_t0  = phddd * ttc_t0
c
c---- enthalpy residual
res1 = (alf*(t0 + bta*t0**2) + p0*pi/tau*phd   )/z0 - h0
r1_p0 = (          pi/tau*phd   )/z0
r1_t0 = (alf*(1.0+ bta*t0*2.) + p0*pi/tau*phd_t0)/z0
c
c---- entropy residual
res2 = alf*log(t0) + 2.0*alf*bta*t0
&      - p0*pi*( t0*phd   *ttc_t0 + ph   ) - log(p0) - s
r2_p0 = - pi*( t0*phd   *ttc_t0 + ph   ) - 1.0/p0
r2_t0 = alf/t0      + 2.0*alf*bta
&      - p0*pi*(   phd   *ttc_t0 + ph_t0
&                + t0*phd_t0*ttc_t0
&                + t0*phd   *ttc_tt0   )
c
c---- setup and invert Jacobian matrix
a(1,1) = r1_t0
a(1,2) = r1_p0
a(2,1) = r2_t0
a(2,2) = r2_p0
c
detinv = 1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
ai(1,1) = a(2,2)*detinv
ai(2,2) = a(1,1)*detinv
ai(1,2) = -a(1,2)*detinv
ai(2,1) = -a(2,1)*detinv
c
c---- set Newton variables
dt = -(ai(1,1)*res1 + ai(1,2)*res2)
dp = -(ai(2,1)*res1 + ai(2,2)*res2)
c
rlx = 1.0
if(rlx*dp .gt. 2.5*p0) rlx = 2.5*p0/dp
if(rlx*dp .lt. -.8*p0) rlx = -.8*p0/dp
if(rlx*dt .gt. 2.5*t0) rlx = 2.5*t0/dt
if(rlx*dt .lt. -.8*t0) rlx = -.8*t0/dt
c
c---- update variables

```

```

    p0 = p0 + rlx*dp
    t0 = t0 + rlx*dt
c
c---- convergence check
    if(abs(dp/p0) .le. eps .and. abs(dt/t0) .le. eps) go to 2
c
    100 continue
c
    write(*,*) 'NONSTAG: Convergence failure.'
    write(*,*) 'dp dT      :',dp, dt
    write(*,*) 'po To h r:',p0,t0,h,r
c
    2 continue
c
c---- set residual derivatives wrt (s,h0)
    r1_s = 0.
    r2_s = -1.0
    r1_h = -1.0
    r2_h = 0.
c
    b(1,1) = r1_s
    b(1,2) = r1_h
    b(2,1) = r2_s
    b(2,2) = r2_h
c
c---- set (t0,p0) derivatives wrt (s,h0)
    t0_s = -(ai(1,1)*b(1,1) + ai(1,2)*b(2,1))
ccc   t0_h0 = -(ai(1,1)*b(1,2) + ai(1,2)*b(2,2))
    p0_s = -(ai(2,1)*b(1,1) + ai(2,2)*b(2,1))
ccc   p0_h0 = -(ai(2,1)*b(1,2) + ai(2,2)*b(2,2))
c
c---- convert derivatives wrt (s,h0) to wrt (p,t,h0)
    t0_t = t0_s*s_t
    t0_p = t0_s*s_p
    p0_t = p0_s*s_t
    p0_p = p0_s*s_p
c
c
c---- set stagnation density r0 and derivatives wrt (p0,t0)
    zz = z(p0,t0)
    zz_p = z_p(p0,t0)
    zz_t = z_t(p0,t0)
c
    r0 = z0/zz      * p0/t0
    r0_z = -z0/zz**2 * p0/t0
c
    r0_p0 = r0_z*zz_p + z0/(zz*t0)
    r0_t0 = r0_z*zz_t - z0*p0/(zz*t0**2)
c
c---- convert derivatives from wrt (p0,t0) to wrt (p,t,h0)
    r0_p = r0_p0*p0_p + r0_t0*t0_p
    r0_t = r0_p0*p0_t + r0_t0*t0_t
ccc   r0_h0 = r0_p0*p0_h0 + r0_t0*t0_h0
c

```

```

c---- convert derivatives from wrt (p,t) to wrt (r,q,h0)
r0_r = r0_p*p_r + r0_t*t_r
r0_q = (r0_p*p_h + r0_t*t_h)*h_q
p0_r = p0_p*p_r + p0_t*t_r
p0_q = (p0_p*p_h + p0_t*t_h)*h_q
c
ccc r0_h0 = (r0_p*p_h + r0_t*t_h)*h_h0 + r0_h0
ccc p0_h0 = (p0_p*p_h + p0_t*t_h)*h_h0 + p0_h0
c
return
end

```

```

real*4 function phi(ttc)
implicit real*4(a-h,m,o-z)

```

```

-----
c Returns function phi used in non-ideality parameter
c Z = 1 + pi*phi(ttc)
-----

```

```

common /nonfit/
& c2, c1, c0
c
phi = c2*ttc**2 + c1*ttc + c0
c
return
end

```

```

real*4 function phid(ttc)
implicit real*4(a-h,m,o-z)
common /nonfit/
& c2, c1, c0
c
phid = 2.*c2*ttc + c1
c
return
end

```

```

real*4 function phidd(ttc)
implicit real*4(a-h,m,o-z)
common /nonfit/
& c2, c1, c0
c
phidd = 2.*c2
c
return
end

```

```

real*4 function phidd(ttc)
implicit real*4(a-h,m,o-z)
common /nonfit/
& c2, c1, c0
c

```

```

    phidd = 0.
c
    return
end

      subroutine hgent(h0,r,q, s)
c-----
c   Returns entropy s from input variables h0,r,q
c-----
      common /nongas/
&         alf, bta, pi, tau, z0
      common /nonfit/
&         c2, c1, c0
c
      h   = h0 - .5*q**2
c
c----- set input pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
&               t,t_r,t_h,t_rr,t_hh,t_rh)
c
      ttc = 1./(tau*t)
      ttc_t = -1./(tau*t**2)
c
      ph   = phi(ttc)
      phd  = phid(ttc)
c
      s   = alf*log(t) + 2.0*alf*bta*t
&       - p*pi*(t*phd*ttc_t + ph) - log(p)
c
      return
end

      subroutine nongamv(h0,r,q, gam,gam_r,gam_q)
c-----
c   Returns "equivalent" gamma for BL density profile
c-----
      common /nongas/
&         alf, bta, pi, tau, z0
      common /nonfit/
&         c2, c1, c0
c
c----- set static enthalpy
      h   = h0 - 0.5*q**2
      h_q =      -q
c
c----- set pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
&               t,t_r,t_h,t_rr,t_hh,t_rh)
c
c----- set speed of sound squared:  a^2 = dp/dr (at constant s)
      asq = p_r / (1. - p_h/r)

```



```

asq_r = p_rr / (1. - p_h/r)
& - p_r / (1. - p_h/r)**2 *(p_h/r**2 - p_rh/r)
asq_h = p_rh / (1. - p_h/r)
& + p_r / (1. - p_h/r)**2 *p_hh/r
c
c
ttc = 1./(tau*t)
ttc_t = -1./(tau*t**2)
ttc_tt = 2./(tau*t**3)
c
ph = phi(ttc)
phd = phid(ttc)
phdd = phidd(ttc)
phddd = phiddd(ttc)
c
z = 1. + p*pi*ph
z_p = pi*ph
z_t = p*pi*phd*ttc_t
c
cp = ( alf*(1.0 + 2.0*bta*t)
& + p*pi/tau* phdd*ttc_t ) / z0
cp_p = ( pi/tau* phdd*ttc_t ) / z0
cp_t = ( alf*( 2.0*bta )
& + p*pi/tau*(phddd*ttc_t**2 + phdd*ttc_tt) ) / z0
c
zet = h/(cp*t)*(1.0 - p*pi/(t*tau)*phd/z) * z0
zet_h = 1.0/(cp*t)*(1.0 - p*pi/(t*tau)*phd/z) * z0
zet_p = h/(cp*t)*( - pi/(t*tau)*phd/z
& - p*pi/(t*tau)*phd/z*(-z_p/z) ) * z0
& - (zet/cp)*cp_p
zet_t = h/(cp*t)*( - p*pi/(t*tau)*phd/z*(-z_t/z - 1.0/t)
& - p*pi/(t*tau)*phdd*ttc_t/z ) * z0
& - (zet/cp)*cp_t - (zet/t )
c
c
gam = asq/(h*zet) + 1.0
gam_r = asq_r/(h*zet)
gam_h = asq/(h*zet)*(-zet_h/zet - 1.0/h) + asq_h/(h*zet)
gam_p = asq/(h*zet)*(-zet_p/zet)
gam_t = asq/(h*zet)*(-zet_t/zet)
c
gam_h = gam_p*p_h + gam_t*t_h + gam_h
gam_r = gam_p*p_r + gam_t*t_r + gam_r
c
gam_q = gam_h*h_q
c
return
end

subroutine sonic(h0,p0,r0, q,p,r)
c-----
c calculates sonic quantities q,p,r
c from specified sonic quantities h0,p0,r0

```

```

c-----
      implicit real (m)
      data eps / 1.0e-5 /
c
c---- initialize with perfect gas
      gam = r0*h0 / (r0*h0 - p0)
      gm1 = gam - 1.0
c
      q = sqrt(2.0*h0/(2.0/gm1 + 1.0))
c
      trat = 1.0 + 0.5*gm1
      p = p0*trat**(-gam/gm1)
      r = r0*trat**(-1.0/gm1)
c
c---- converge on non-ideal values by forcing M^2 = 1, and pstag = p0
      do 10 iters=1, 15
          call nideal(h0,r,q, p ,p_r ,p_q,
&                  msq,msq_r,msq_q )
          call nonstag(h0,r,q, pstag,psdag_r,psdag_q,
&                  rstag,rstag_r,rstag_q )
          res1 = msq - 1.0
          a11 = msq_r
          a12 = msq_q
c
          res2 = pstag - p0
          a21 = psdag_r
          a22 = psdag_q
c
          detinv = 1.0/(a11*a22 - a12*a21)
          dr = -(res1*a22 - a12 *res2)*detinv
          dq = -(a11 *res2 - res1*a21 )*detinv
c
          dp = p_r*dr + p_q*dq
c
          rlx = 1.0
          if(rlx*dr .gt. 1.5*r) rlx = 1.5*r/dr
          if(rlx*dr .lt. -.6*r) rlx = -.6*r/dr
          if(rlx*dq .gt. 1.5*q) rlx = 1.5*q/dq
          if(rlx*dq .lt. -.6*q) rlx = -.6*q/dq
c
          r = r + rlx*dr
          q = q + rlx*dq
          p = p + rlx*dp
c
          dmax = amax1( abs(dr)/r , abs(dq)/q )
c
          if(dmax .lt. eps) go to 11
c
10      continue
      write(*,*) 'sonic: convergence failed. dmax =', dmax
11      continue
c
      return
      end ! sonic

```

## Bibliography

- [1] J. H. McMasters, W. H. Roberts, and F. M. Payne. Recent air-freon tests of a transport airplane in high lift configurations. AIAA-88-2034, 1988.
- [2] H. W. Liepmann and A. Roshko. *Elements of Gasdynamics*. Wiley, New York, 1957.
- [3] B. Wagner and W. Schmidt. Theoretical investigations of real gas effects in cryogenic wind tunnels. *AIAA Journal*, 16(6), June 1978.
- [4] M. Drela and M. B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10), Oct 1987.
- [5] W. Anderson. A numerical study on the use of sulfur hexafluoride as a test gas for wind tunnels. AIAA-90-1421, 1990.
- [6] O. Coufal. Thermodynamic properties of sulphur hexafluoride in temperature range 298.15 – 30,000 K and pressure range .101325 – 2 MPa. *ACTA Technica CSAV*, 31, 1986.